Challenges in Scalable Clusters For Technical Computing

October 11, 2001

Bill Camp
Director of Computation, Computers & Mathematics Center

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy under contract DE-AC04-94AL85000.

Sandia National Laboratories

Taking Stock on Cluster-based virtual supercomputing

- Challenges in:
  - Design
  - Integration
  - Management
  - Use

Original(199) Goal for Cplant^{TM}

- Scalable, Reliable, Evolving Virtual Supercomputers for a world with no HPC vendors.

Strategies:

- build on commodity
- leverage Open Source (eg Linux)
- Add to commodity selectively
- Provide look and feel of scalable supercomputers

Underlying Question:
(with 3+ years under our belt)

Is cluster-based supercomputing a viable *general purpose* **solution at the highest end?**

If yes, what is needed to make it succeed?

If no, where do we go from here?

What’s Important?

USR:

- Usability
- Scalability
- Reliability

Context - Very Large Parallel Computer Systems

Usability - Required Functionality Only

Scalability - Full System Hardware and System Software

Reliability - Hardware and System Software

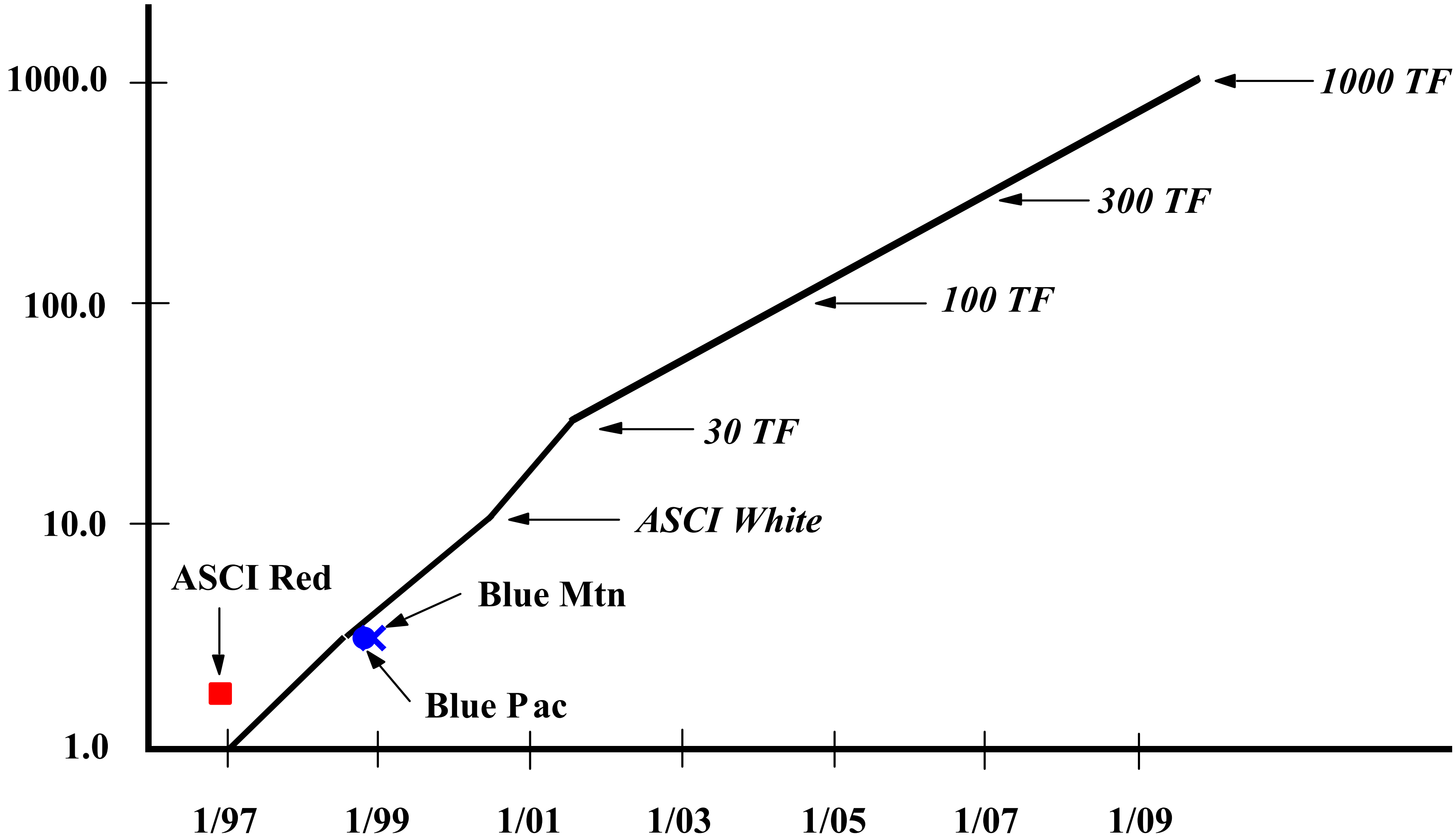
USR poses Computer System Requirements:

GOAL:

Computer Architecture
for
100 TOPs and Beyond

Sandia National Laboratories

Extended "ASCI" Curve



Sandia National Laboratories

ASCI Curve implies:

Computer Systems with Thousands (~20,000) of Processors.

Typical calculations require a large fraction of the total machine resources for a hundred or more hours.

Some Examples of problems that need really large scale computing capabilities:

- Micro and Macro Weather Simulations
- Global Climate Simulations
- Material Ageing
- Drug Design
- Human Biology - Brain, Circulatory System, etc.
- Weapons Physics
- Weapons Safety
- Intelligent agent models

Usability

Application Code Support:

Software that supports scalability of the Computer System
- **Math Libraries**
- **MPI Support for Full System Size**
- **Parallel I/O Library**
- **Compilers**

Tools that Scale to the Full Size of the Computer System
- **Debuggers**
- **Performance Monitors**

Full OS support at the user interface

Scalability

Hardware:

System Hardware Performance increases linearly with the number of processors to the full computer system size - Scaled Speedup.

- Avoidance of Hardware bottlenecks
- Communication Network performance
- I/O System

Machine must be able to support ~20,000 processors operating as a single system.

Scalability

System Software:

System Software Performance scales nearly perfectly with the number of processors to the full size of the computer (~20,000 processors). This means that System Software time (overhead) remains nearly constant with the size of the system or scales at most logarithmically with the system size.

- Full re-boot time scales logarithmically with the system size.
- Job loading is logarithmic with the number of processors.
- Parallel I/O performance doesn't depend on how many PEs are doing I/O
- Communication Network software must be scalable.
No connection-based protocols.
Message buffer space independent of # of processors.

Compute node OS gets out of the way of the application.

Sandia National Laboratories

Scaling Analysis

Consider three application parallel efficiencies on 1000 processors. What is the most productive way to increase overall application performance?

Case 1: 90% Parallel Efficiency

10X faster processor yields ~5X application code speedup

Cut parallel inefficiency by 10X makes 5% increase in speed

Case 2: 50% Parallel Efficiency

10X faster processor yields <2X application code speedup

Cut parallel inefficiency by 10X makes ~2X increase in speed

Case 3: 10% Parallel Efficiency

10X faster processor yields ~10% application code speedup

Cut parallel inefficiency by 10X makes ~9X increase in speed

System Scalability Driven Requirements

Overall System Scalability - Complex scientific applications such as radiation transport should achieve scaled parallel efficiencies greater than 70% on the full system (~20,000 processors).

- This implies the need for excellent interconnect performance, hardware and software.
- Overlap of communication and computation is difficult to achieve for most scientific codes.

Overall System Reliability - The usefulness of the system is strongly dependent on the time between interrupts.

- Ratio of calculation time to time spent checkpointing should be ~20 to 1 to make good progress.
- 100 hour MTBI is desirable

What makes a computer scalable

- Balance in the hardware:
  - Memory BW must match CPU speed

    Ideally 24 Bytes/flop (never yet done)
  - Ewald's Folk Theorem:

    Real Speed < Min[(CPU Speed, Mem.BW)/4]
  - Communications speed must match CPU speed
  - I/O must match CPU speeds
- Scalable System SW(OS and Libraries)
- Scalable Applications

What doesn't help scalability

- Shared Memory:
    - Cache Coherency actually hurts scalability for large #'s of CPUs
    - Shared memory programming methods (eg threads) do not scale to large #'s of CPUs
- Virtual Memory in App's space-- "Paging to where?"

Let’s Compare Balance In Parallel Systems

Machine	Node Speed Rating(MFlops)	Link BW (Mbytes/s)	Ratio (Bytes/flop)
ASCI RED	400	800(533)	2(1.33)
T3E	1200	1200	1
ASCI RED**	666	800(533)	(1.2)0.67
Antarctica	932	140	0.15
Blue Mtn*	500	800	1.6
BlueMtn**	64000	1200 (9600*)	0.02 (0.16)
Blue Pacific	2650	300 (132)	0.11 (0.05)
White	24000	2000	0.083
30T*	2500	650	0.26
30T**	80000	400	0.05

Sandia National Laboratories

Why is Comm's the Killer Concern?

People have been led to think that Amdahl's Law limits the scalability of parallel computation

In Theory it does, in actuality it doesn't.

Why?

Amdahl's Law

$$S_{Amdahl}(N) = [1 + f_s]/[1/N + f_s]$$

where S is the speedup on N processors and f_s is the serial (non-parallelizable) fraction of the work to be done.

Amdahl says that in the limit of an infinite number of processors, S cannot exceed $[1+ f_s]/ f_s$. So, for example if $f_s = 0.01$, S cannot be greater than 101 no matter how many processors are used.

Amdahl's Law

Example:

How big can f_s be if we want to achieve a speedup pf 8,000 on 10,000 processors (80% parallel efficiency)?

Answer:

f_s must be less than 0.000025 !

Amdahl’s Law

The good news is that contrary to Amdahl’s expectation, we can routinely do this well or better!

The bad news is that Amdahl neglected the overhead due to communications.

A more REAListic Law

The actual scaled speedup is more like

$$S(N) \sim S_{Amdahl}(N)/[1 + f_{comm} \times R_{p/c}],$$

where f_{comm} is the fraction of work devoted to communications and $R_{p/c}$ is the ratio of processor speed to communications speed.

REAL Law Implications

$S_{real}(N) / S_{Amdahl}(N)$

Let's consider three cases on two computers:

the two computers are identical except that one has an $R_{p/c}$ of 1 and the second an $R_{p/c}$ of 0.05

The three cases are f_{comm} = 0.01, 0.05 and 0.10

REAL Law Implications

$S(N) \,/\, S_{Amdahl}(N)$

f_{comm} / $R_{p/c}$	0.01	0.05	0.10	
1.0	0.99	0.95	0.9	
0.05	0.83	0.50	0.33	

Sandia National Laboratories

Bottom line:

A well-balanced architecture is nearly insensitive to communications overhead

By contrast a system with weak communications can lose over half its power for applications in which communications is important

Applications Scalability Driven Requirements

High Performance Machine Interconnect

Bandwidth - at least 1 B/F

MPI Latency (ping-pong divided by 2) - ~3000 CPU clocks

System Software Scalability

- No large SMPs-- N^2 cost and overhead scaling

- No connection based networks - N^2 scaling

- Source based routing

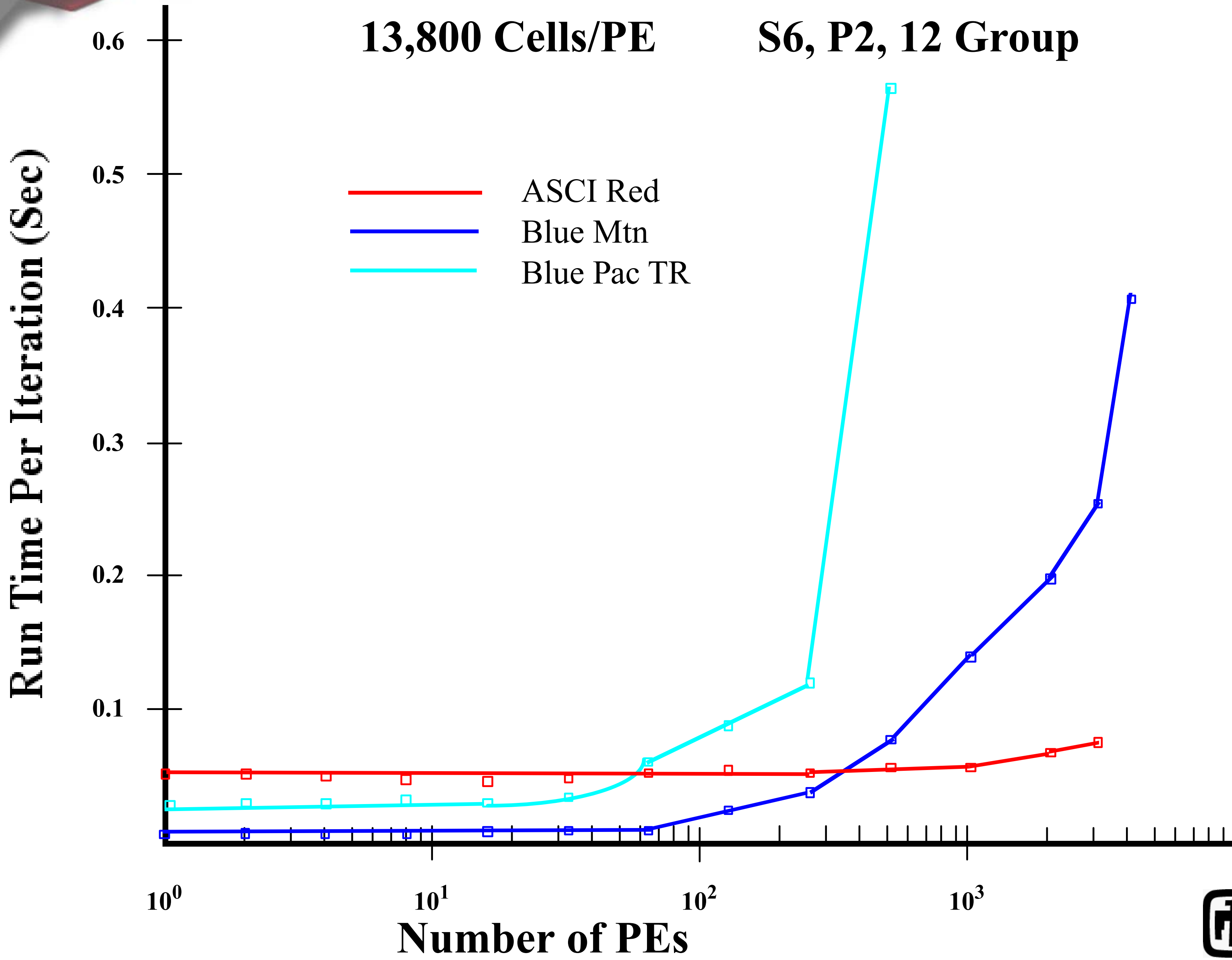
- Compute Node OS - No time sharing of nodes, No compute node paging, No sockets, No spurious demons, Minimize number of OS initiated interrupts.

Keep it simple

Overall System Reliability

System MTBI of 50 hrs or more to get useful work done

Parallel S_n Neutronics



Sandia National Laboratories

Conclusion:

For most large scientific and engineering applications the performance is determined by parallel scalability and not the speed of individual CPUs. There must be balance between processor, interconnect, and I/O performance to achieve overall performance.

To date, only a few tightly-coupled, parallel computer systems have been able to demonstrate a high level of scalability on a broad set of scientific and engineering applications. No clusters yet have. Cplant^{TM} shows promise.

Sandia National Laboratories

Reliability for Scientific and Engineering Applications

What is Reliability:

- High Mean Time Between Interrupts for hardware and system software
- High Mean Time Between errors/failures that affect users

What it is not:

- High availability

How to Get Reliability: System Software

Partitioned Operating System (OS)

- Service Partition - Full function OS
- I/O Partition - Full function OS
- Compute Partition - Light Weight Kernel OS
- System Partition - System control functions
- Provide only needed functionality for each partition.

System Software Adaptation

- Automatic OS re-boots on OS failures
- Automatic system reconfiguration for hardware failures

Keep it Simple

How to Get Reliability-- Hardware

A full system approach - Machine must be looked at as a whole and not a bunch of separate parts or sub-systems.

Hardware

- Redundant Components
- Error Correction
- Hot Spares
- Integrated Full System Monitoring and Scalable Diagnostics
- Preventive Maintenance

Is a 50 Hour MTBI Possible?

ASCI Red Experience in 1999

Hardware MTBI - > 900 hours

System Software MTBI - > 40 hours

ASCI Red has over 9000 processors

~4 hours Preventive Maintenance is performed per week

Integrated full system monitoring capability

Almost all unscheduled interrupts occur as a result of OSF/1 failures

(We believe that the software MTBI would be much better if Intel had remained in the supercomputer business.)

Sandia National Laboratories

So, what about Cplant^{TM}?

So, what about Cplant^{TM}?

Cplant^{TM} is growing and thriving:

Currently around 2.5 TF total

One part of Antarctica with 1524 processors achieved over 750 Gflops on MP-Linpack.

We are aiming for 1 TF on MP-Linpack this year!

So, what about Cplant^{TM}?

The Cplant^{TM} System has been released under GPL

We have also given a non-exclusive commercial license to one company--USI

Others are interested.

So, what about Cplant^{TM}?

Cplant^{TM} has demonstrated extremely competitive applications performance for a wide variety of problems out to several hundred processors.

So, what about Cplant^{TM}?

However, ...

However

- Integration remains an issue
- Debugging of HW and SW is non-trivial
- The network was too lean in early versions
- Reliability is not up to that of integrated supercomputers like ASCI RED

  (and may never be at those scales)

So, what about Cplant^{TM}?

We have responded:

- Created a systematic approach to integration-- a department level team
- Created a rigorous code engineering process, including very disciplined testing
- Made a much better network in Antarctica
- ...

At a cost:

We take months to integrate systems;

We have moved to a less frequent growth strategy;

We have duplicated much of the value added by a commercial company
(the extreme Linux community has not yet emerged to provide the development advantages we had hoped for)

...

Current Bottom Line

Cplant^{TM} is a cost effective solution…

- scalability competitive with most current offerings
- reliability will be similar to large ASCI machines (other than RED)
- it provides a foundation for an Open-Source approach to very-high end supercomputing.
- It is much more like RED or the T3E than it is like a Beowulf cluster.

Cplant^{TM} Status

- Big chunks of Antarctica and all of Alaska are in "general availability" mode
- For several months 50--70% of available cycles on these clusters have been consumed by production jobs.
- Siberia has been dismantled and is being reconfigured as part of Antarctica
- Release 1.0 marked the start of true production availability(April'01)
- ALASKA will soon retire

Sandia National Laboratories

Cplant^{TM} Applications Work In Progress

- **CTH**
  - **3D Eulerian shock physics**
- **ALEGRA**
  - **3D arbitrary Lagrangian-Eulerian solid dynamics**
- **GILA**
  - **Unstructured low-speed flow solver**
- **MPQuest**
  - **Quantum electronic structures**
- **SALVO**
  - **3D seismic imaging**
- **LADERA**
  - **Dual control volume grand canonical MD simulation**
- **Parallel MESA**
  - **Parallel OpenGL**
- **Xpatch**
  - **Electromagnetism**
- **RSM/TEMPRA**
  - **Weapon safety assessment**
- **ITS**
  - **Coupled Electron/Photon Monte Carlo Transport**
- **TRAMONTO**
  - **3D density functional theory for inhomogeneous fluids**
- **CEDAR**
  - **Genetic algorithms**

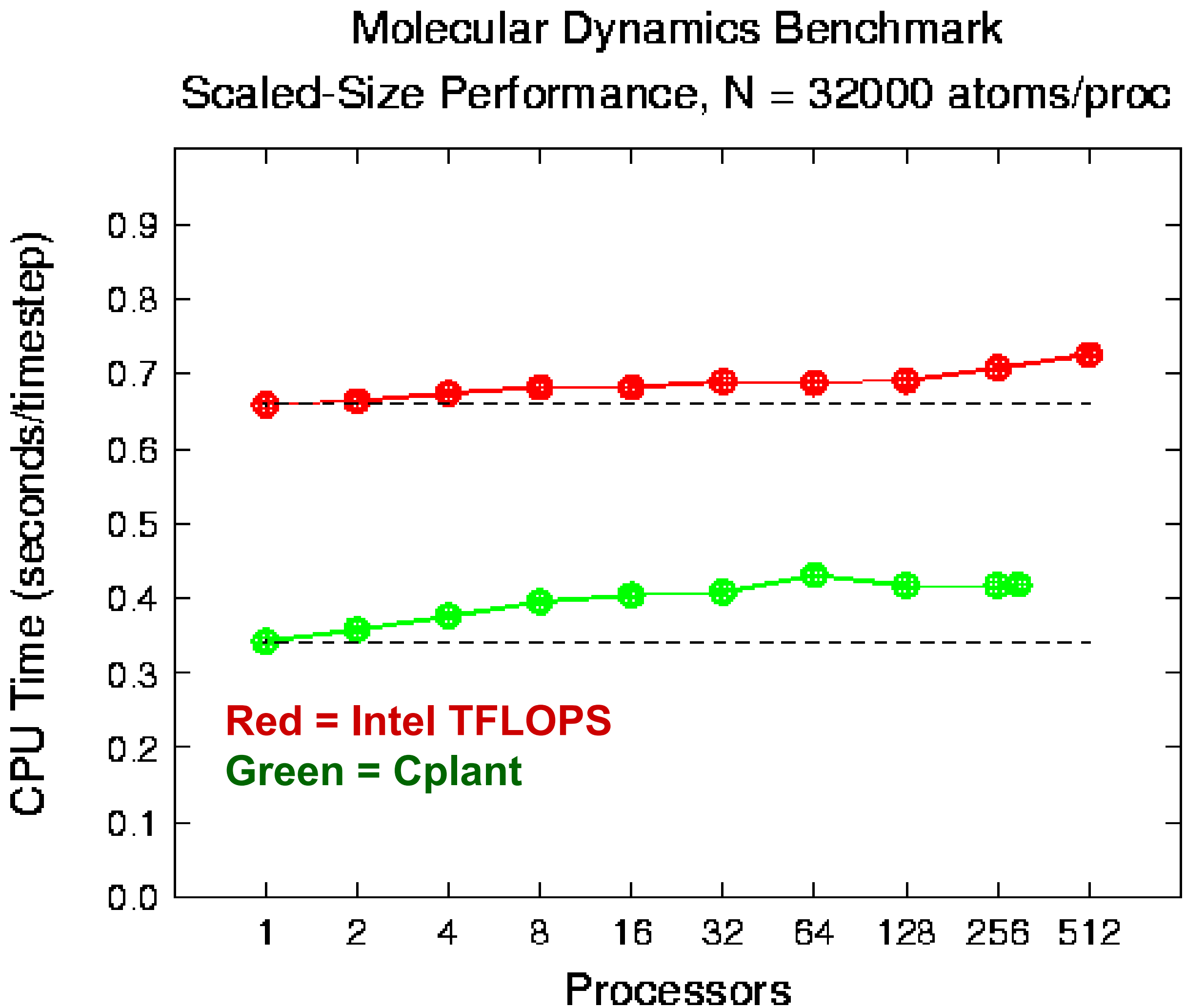
Sandia National Laboratories

Cplant^{TM} Applications Work In Progress

- **AZTEC**
  - Iterative sparse linear solver
- **DAVINCI**
  - 3D charge transport simulation
- **SALINAS**
  - Finite element modal analysis for linear structural dynamics
- **TORTILLA**
  - Mathematical and computational methods for protein folding
- **EIGER**
- **DAKOTA**
  - Analysis kit for optimization
- **PRONTO**
  - Numerical methods for transient solid dynamics
- **SnRAD**
  - Radiation transport solver
- **ZOLTAN**
  - Dynamic load balancing
- **MPSALSA**
  - Numerical methods for simulation of chemically reacting flows

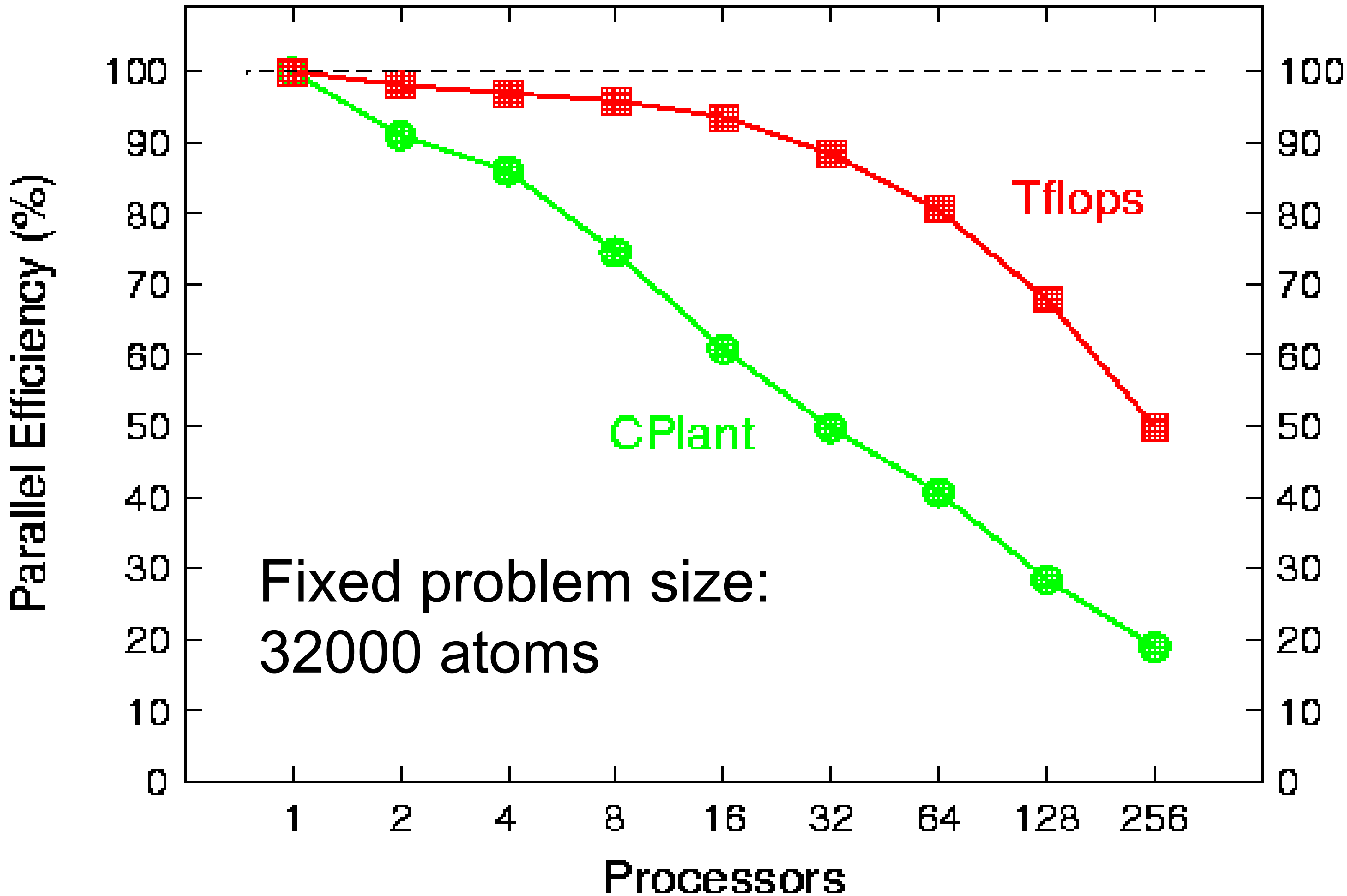
http://www.cs.sandia.gov/cplant/apps

Cplant^{TM} Performance

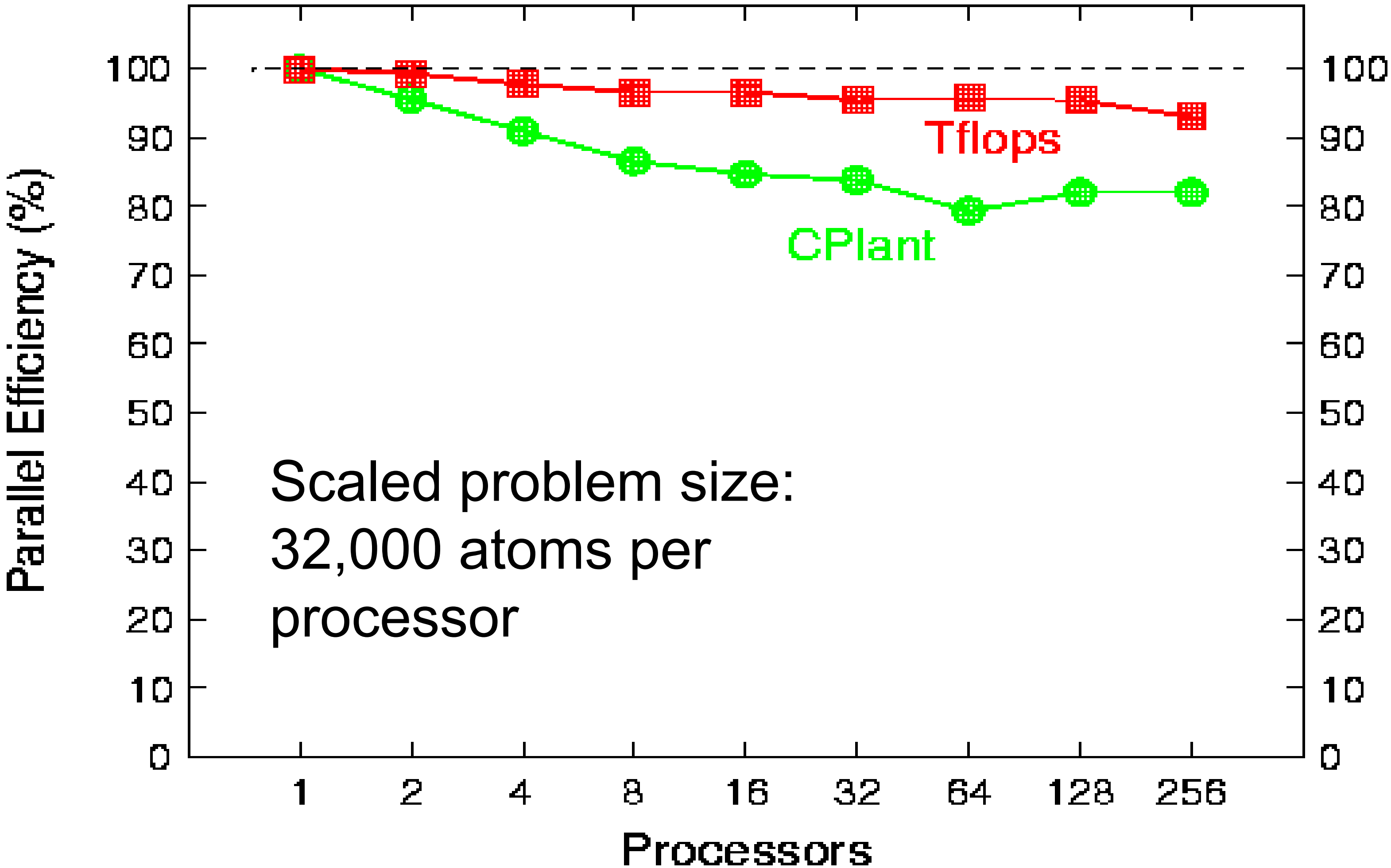


Sandia National Laboratories

L-J Liquid M.-D. Benchmark

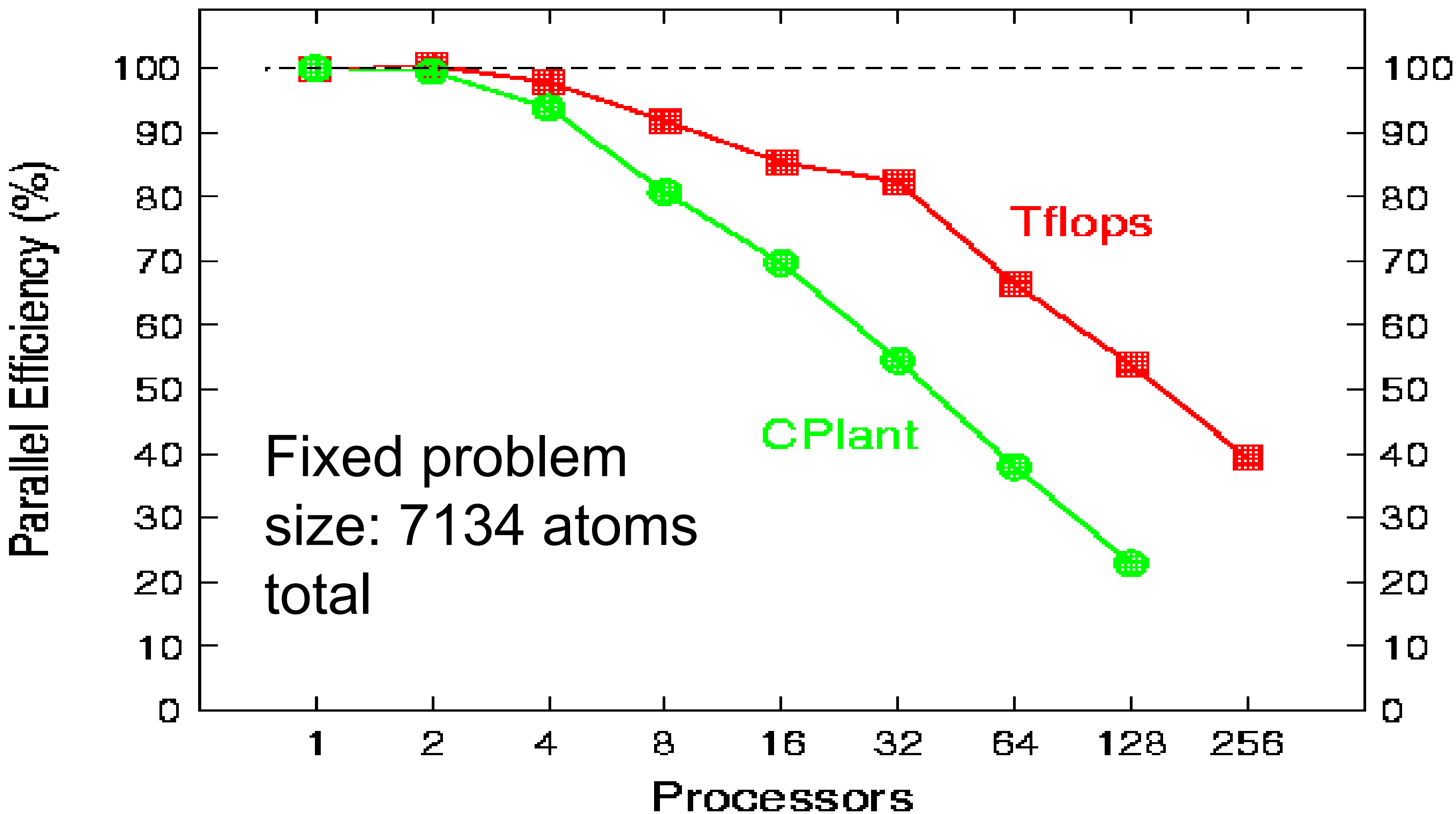


L-J Liquid M.-D. Benchmark

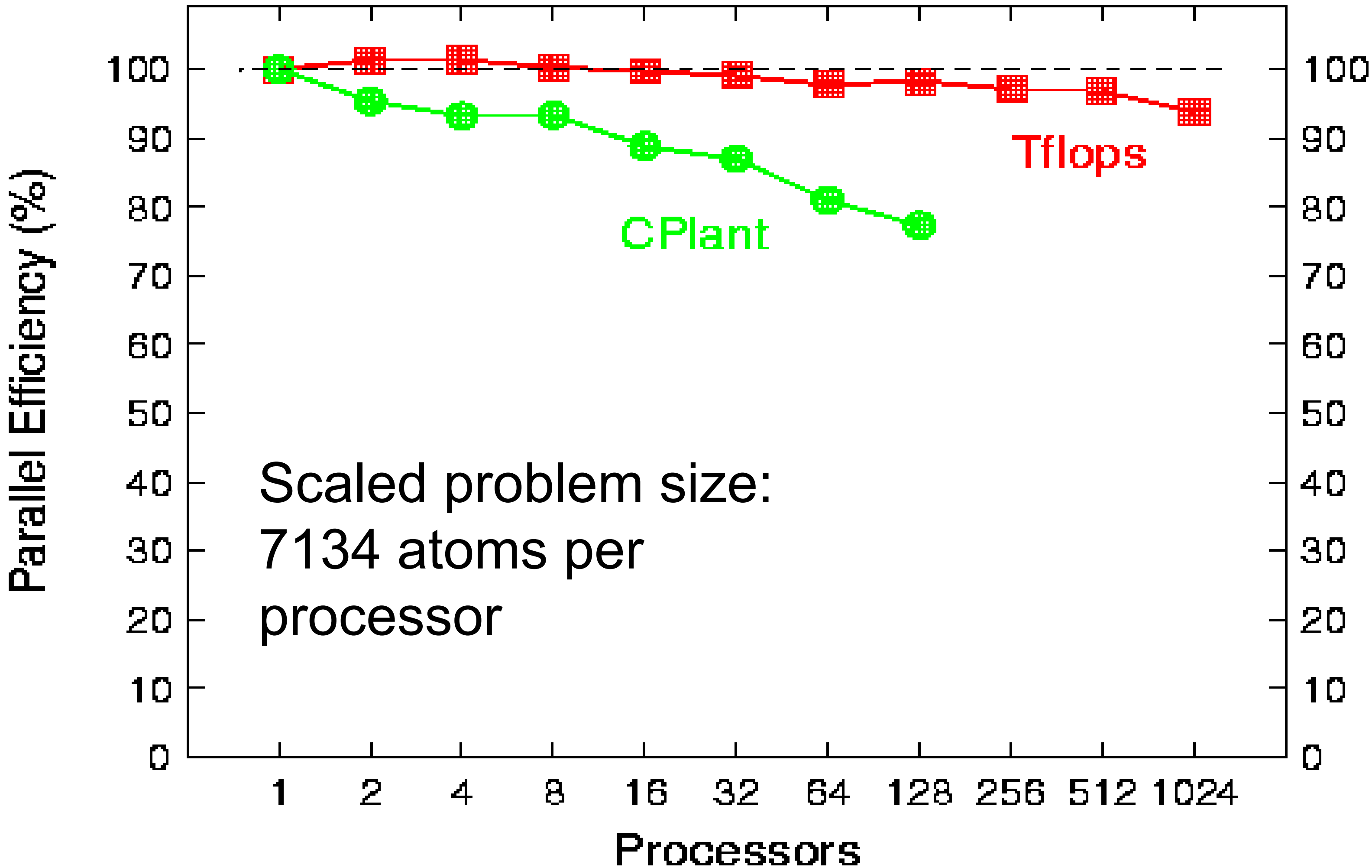


Sandia National Laboratories

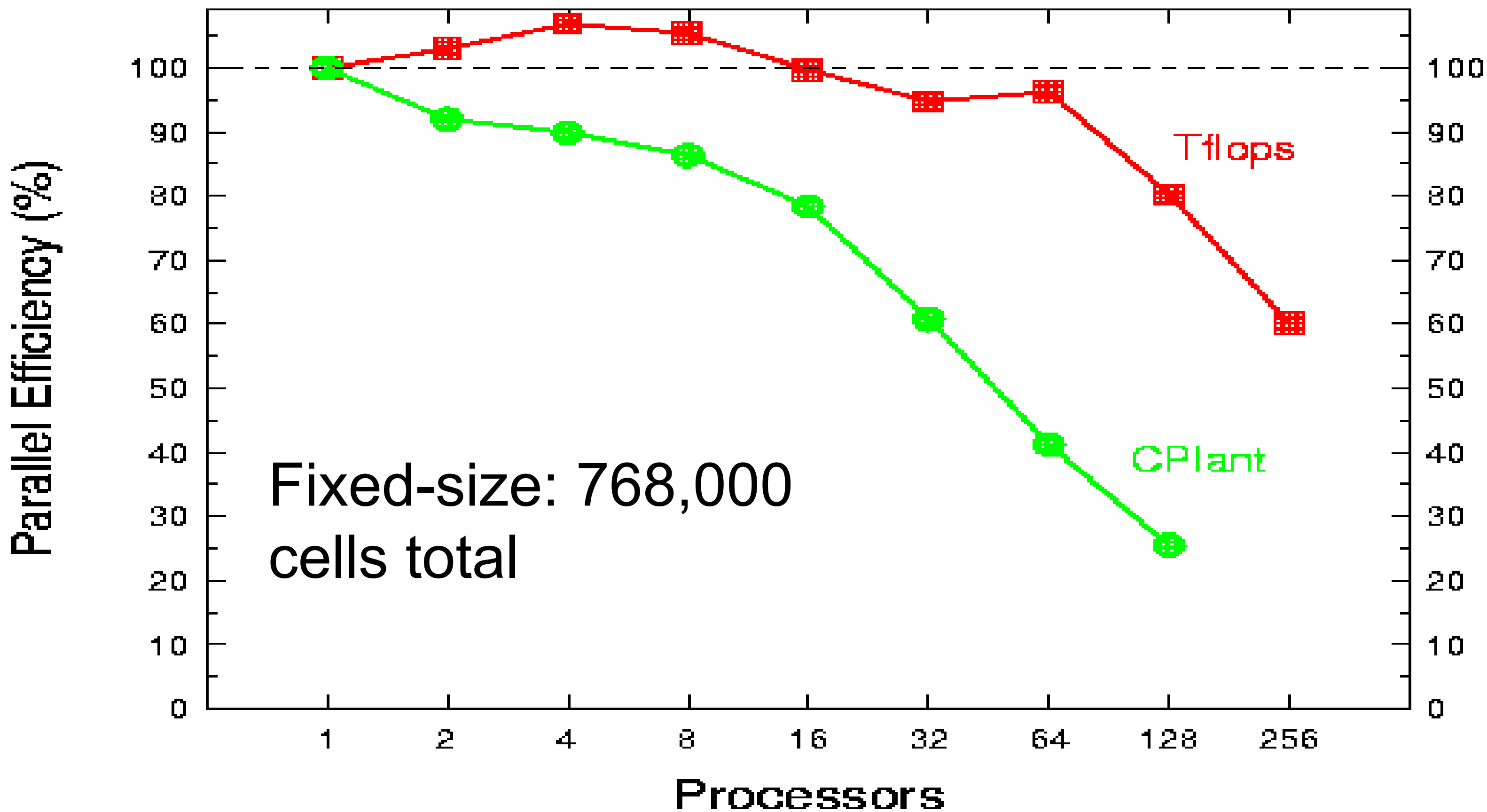
LAMMPS MD Simulation of a solvated lipid bi-layer



LAMMPS MD Simulation of a solvated lipid bi-layer

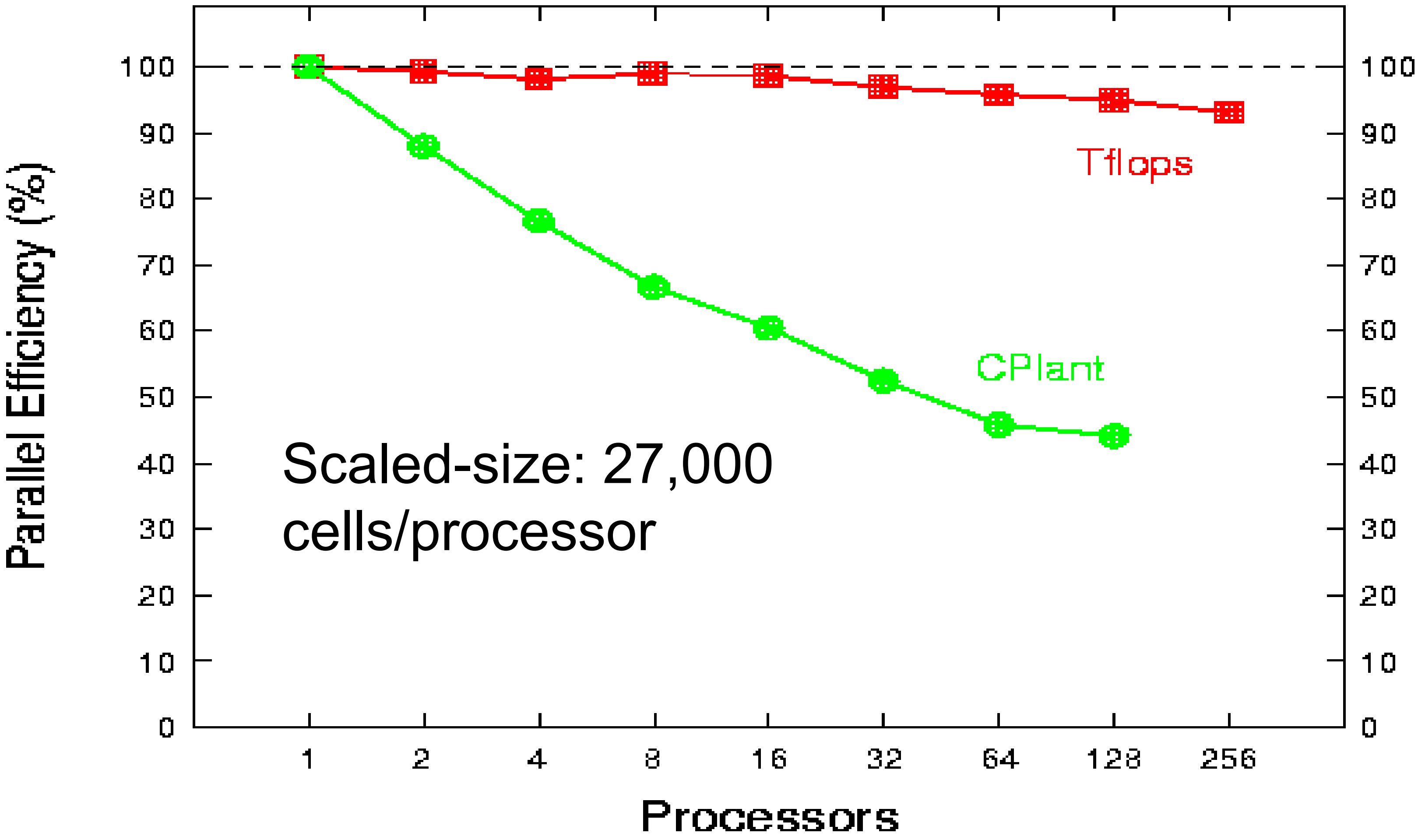


QuickSilver EM simulation for a travelling-wave pulse

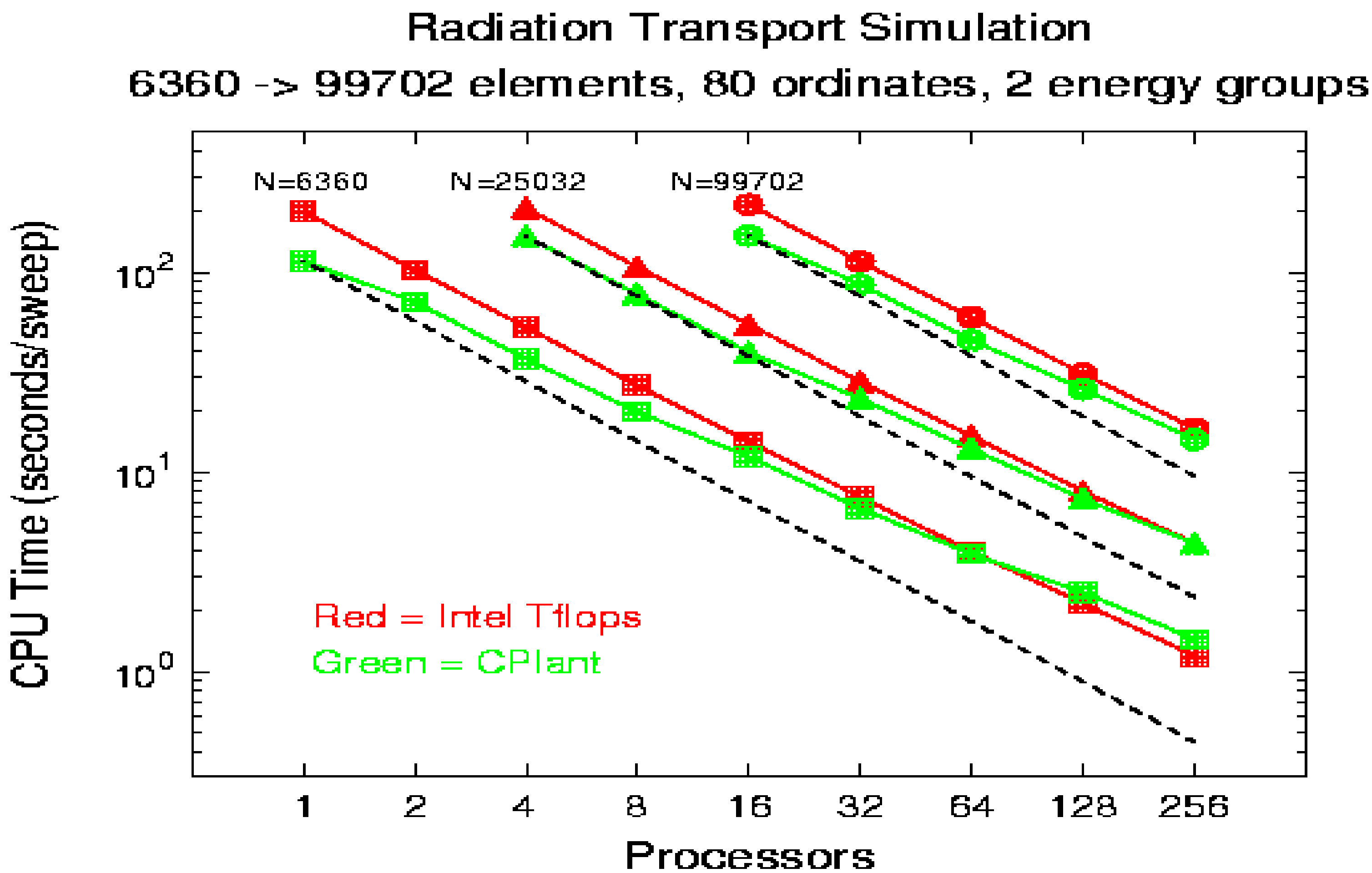


Sandia National Laboratories

QuickSilver EM simulation for a travelling-wave pulse



3-d Unstructured Radiation Transport problem



Sandia National Laboratories

Parallel S_n Neutronics